

# AN INVERSE KINEMATICS ANALYSIS OF SPACE STATION REMOTE MANIPULATOR SYSTEM (SSRMS) USING GENETIC ALGORITHMS

SATYENDRA JALADI & T. E. RAO

Department of Mechanical Engineering, K L E F (Deemed to be University),  
Vaddeswaram, Guntur, Andhra Pradesh, India

## ABSTRACT

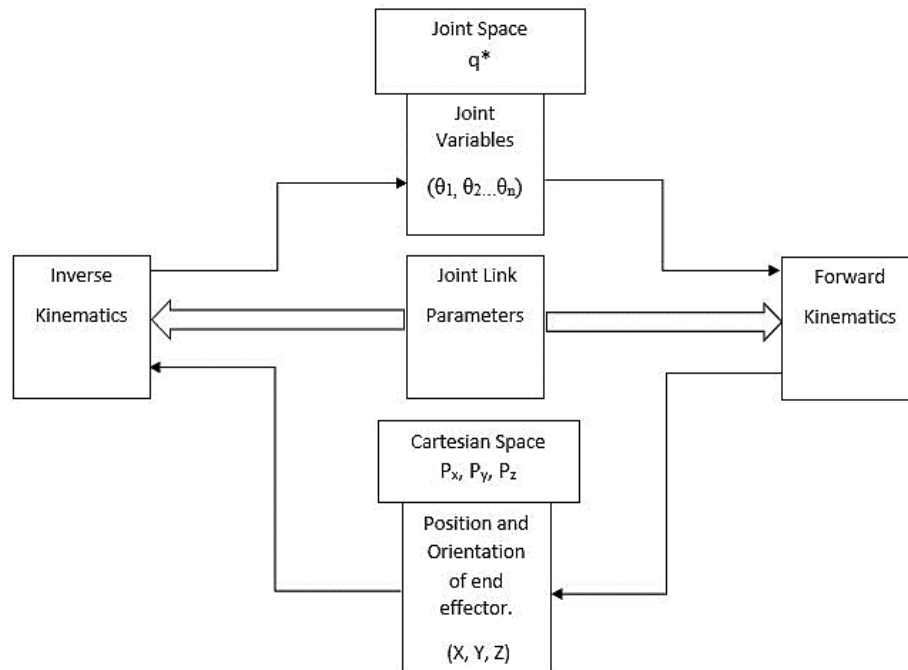
*The importance of Robot Arms (Serial Manipulators) is increasing rapidly in the last few years in space, industrial and medical applications. An important part of Robot Arms is to achieve desired pose (position and orientation), in order to accomplish this very good knowledge of inverse kinematics is needed. Getting a solution to Inverse kinematics of the Robot Arms has been considered as a mature problem which is thoroughly researched and is also on the focus of various research and developments in Robotics field. SSRMS (a redundant manipulator) has total 7 revolute joints. They can be structured as shoulder joints, elbow joints and wrist joints (end effector). This paper presents biologically-inspired Genetic Algorithms (optimization) approach for solving Inverse kinematics problem of SSRMS. In this paper GA solver from MATLAB's (optimization toolbox) is used to find the joint angles of SSRMS robot, solver parameters such as population size, elite count, crossover fraction, etc. are selected to achieve maximum performance resulting in optimum joint variables.*

**KEYWORDS:** Genetic Algorithm, GA, Inverse Kinematics, Serial Manipulators, SSRMS & Optimization

**Received:** Nov 22, 2018; **Accepted:** Dec 12, 2018; **Published:** Jan 05, 2019; **Paper Id.:** IJMPERDFEB201921

## 1. INTRODUCTION

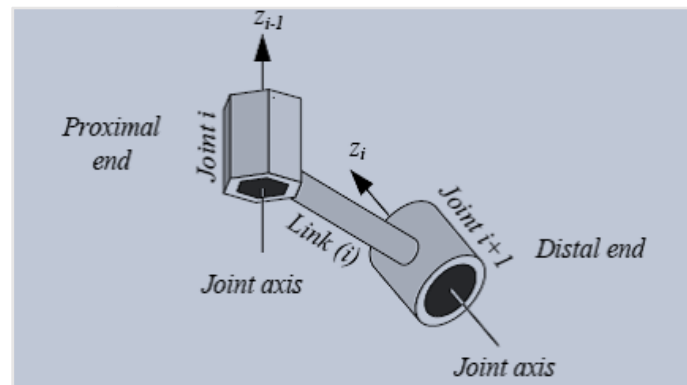
Robotic systems will play an increasingly important role in future space activities, such as repairing, refueling, re-orbiting, assembling and upgrading spacecraft [1-5]. ASSRMS (Canadarm2) has seven joints, each joint has a large movement range of  $\pm 270^\circ$  [6] and links have offset lengths to avoid any mechanical interference. The joints are subjected to the functions of controllers, sensors and actuators. Resolving Position level Inverse kinematics of SSRMS is not similar to solving Inverse kinematics of Industrial manipulator which has 6 degrees of freedom, where a spherical wrist (three adjacent joints) is attached to the end effector. The joints of SSRMS are structured like this. Shoulders are connected by three joints, two large boom like links (lower and upper arms) are connected by one elbow joint. Three joints connect the Wrist which holds end-effector. In the current approach, the Forward kinematics equations (trigonometry) which will give a target point ( $X_t$ ,  $Y_t$ ,  $Z_t$ ) are written in Fitness Function. Travelling Salesman Problem (TSP) from combinatorial optimization, domain, is considered for minimizing the position error. With the given set values (constraints) GA solver (MATLAB's optimization toolbox) returns the optimum joint angles for SSRMS. The relationship between inverse kinematics and direct/forward kinematics can be seen in Figure 1.



**Figure 1: Relationship between Forward and Inverse Kinematics**

## 2. FORWARD KINEMATICS

Given joint link parameters of a robot determining its configuration (position and orientation) of every link using rigid motion method are Forward kinematics [7]. Link (i) connected to joint (i) and joint (i+1), to get the kinematic information of all the links by attaching a coordinate frame to each link (i) at joint (j + 1) is known as Denavit-Hartenberg method.



**Figure2:Link(i) Connected to Joint (i) and Joint (i + 1)**

SSRMS end effector pose can be obtained from the Homogeneous Transformation Matrix (T).

$$T = T_1 \times T_2 \times T_3 \times T_4 \times T_5 \times T_6 \times T_7$$

The Forward kinematics can be given by multiplication of individual transformation matrices  $T_1, T_2, T_3, T_4, T_5, T_6, T_7$  are defined in Table 1.

### 3. INVERSE KINEMATICS

Given the configuration of a robot determining joint variables associated with the links is Inverse kinematics (IK).

Finding the elements of vector  $[q_i]$  for the given transformation  ${}^0T_n$ .

$${}^0T_n = {}^0T_1(q_1) \times {}^1T_2(q_2) \times {}^2T_3(q_3) \times \dots \times {}^{n-1}T_n(q_n)$$

$$[q_1, q_2, q_3, \dots, q_n]$$

Where  $n \in [1, \dots, j]$

Inverse kinematics problem can be decoupled into rotation matrices and translation matrices.

$${}^0T_7 = {}^0T_3 \times {}^3T_6 \times {}^6T_7$$

$${}^0T_7 = \begin{bmatrix} {}^0_3R & 0 \\ 0 & I \end{bmatrix} \times \begin{bmatrix} {}^3_6R & 0 \\ 0 & I \end{bmatrix} \times \begin{bmatrix} I & d_7 \\ 0 & I \end{bmatrix}$$

### 4. CONFIGURATION OF SSRMS MANIPULATOR

SSRMS has total 7 revolute joints, they are

- Shoulder joints(J1J2J3).
- Elbow joint ( J4 ).
- Wrist joints (J5J6J7).

D-H frames, joints and configuration of SSRMS is shown in Figure 3 and Figure 4.

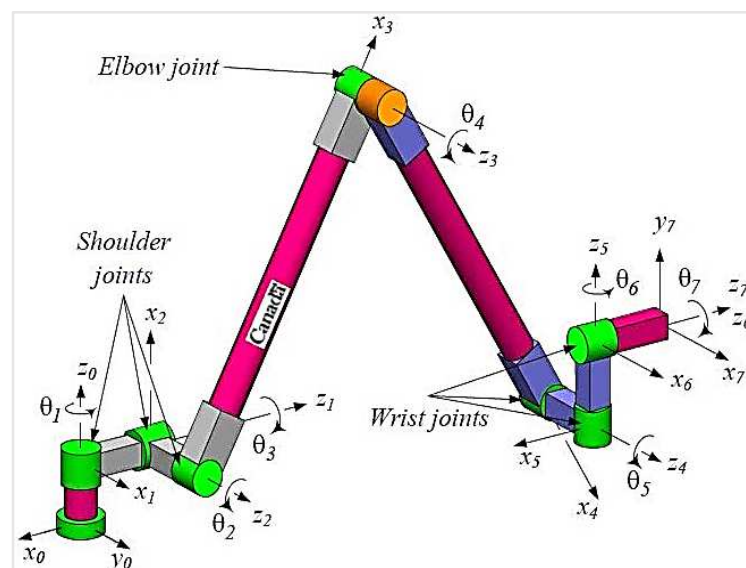
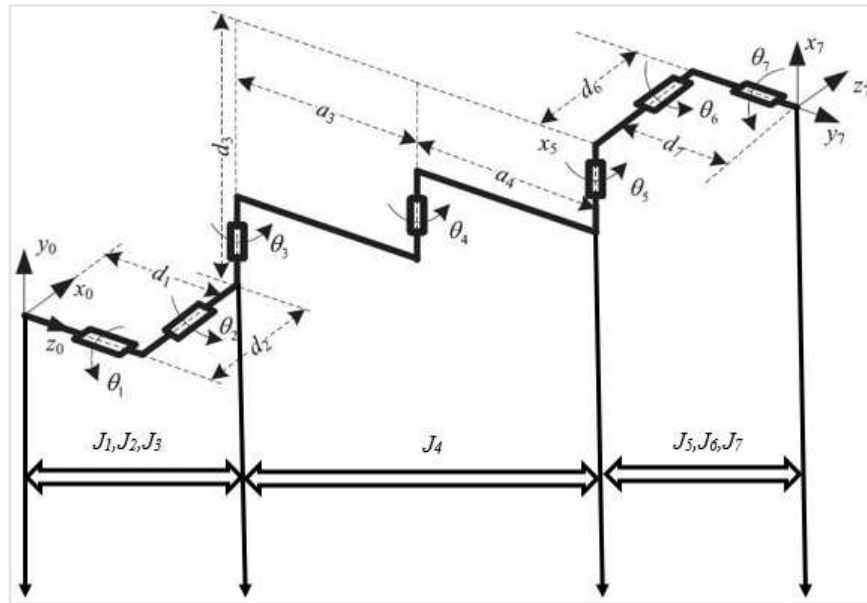


Figure3: CAD Model of SSRMS with 7 Joints and DH-Frames(Base to End-Effector)



**Figure4: Configuration of SSRMS Manipulator  
(Zero Displacement) with 7 Joints and DH-Frames**

**Table 1: SSRMS Forward Kinematics Transformation Matrices**

Rotation Matrices	Translation Matrices
$T_1 = \begin{pmatrix} c(\theta_1) & 0 & -s(\theta_1) & 0 \\ s(\theta_1) & 0 & c(\theta_1) & 0 \\ 0 & -1 & 0 & -d_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	
$T_2 = \begin{pmatrix} c(\theta_2) & 0 & -s(\theta_2) & 0 \\ s(\theta_2) & 0 & c(\theta_2) & 0 \\ 0 & -1 & 0 & d_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	
$T_3 = \begin{pmatrix} c(\theta_3) & -s(\theta_3) & 0 & a_3 \times c(\theta_3) \\ s(\theta_3) & c(\theta_3) & 0 & a_3 \times s(\theta_3) \\ 0 & 0 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	
$T_4 = \begin{pmatrix} c(\theta_4) & -s(\theta_4) & 0 & a_4 \times c(\theta_4) \\ s(\theta_4) & c(\theta_4) & 0 & a_4 \times s(\theta_4) \\ 0 & 0 & 1 & d_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	
$T_5 = \begin{pmatrix} c(\theta_5) & 0 & s(\theta_5) & 0 \\ s(\theta_5) & 0 & -c(\theta_5) & 0 \\ 0 & 1 & 0 & d_5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	

$T_6 = \begin{pmatrix} c(\theta_6) & 0 & -s(\theta_6) & 0 \\ s(\theta_6) & 0 & c(\theta_6) & 0 \\ 0 & -1 & 0 & d_6 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	
	$T_7 = \begin{pmatrix} c(\theta_7) & -s(\theta_7) & 0 & 0 \\ s(\theta_7) & c(\theta_7) & 0 & 0 \\ 0 & 0 & 1 & d_7 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

## 5. GENETIC ALGORITHMS (GA)

GA[8] and TSP is used for joint angle optimization in this study. Since GA is a meta heuristics algorithm, the results do not guarantee an exact optimal solution, but they give near optimal solutions. GA is good at navigating through larger search spaces, where locating a local-minima and applying mutation resulted in a very effective solution. In this study joint angles of SSRMS are derived using the methodologies applied for TSP using GA. If  $(P_x, P_y, P_z)$  is denoted as a known point, the first step is determining the location of the target point  $(X_t, Y_t, Z_t)$  for path planning. Forward kinematics equations written in fitness function will give the target point location, the second step is to achieve global optimum by minimizing the distance between the target and known location. Constraints on joint angles (Lower and Upper bounds) will minimize GA search space, increasing the possibility of obtaining optimum joint angles in a smaller number of generations. The number of generations has been set to 200. When an average change in function fitness value becomes less than the function tolerance  $10e-6$ , GA stops the optimization in this case. Peter Corke's Robotics Toolbox for MATLAB[9] is used to visualize graphical SSRMS robot.

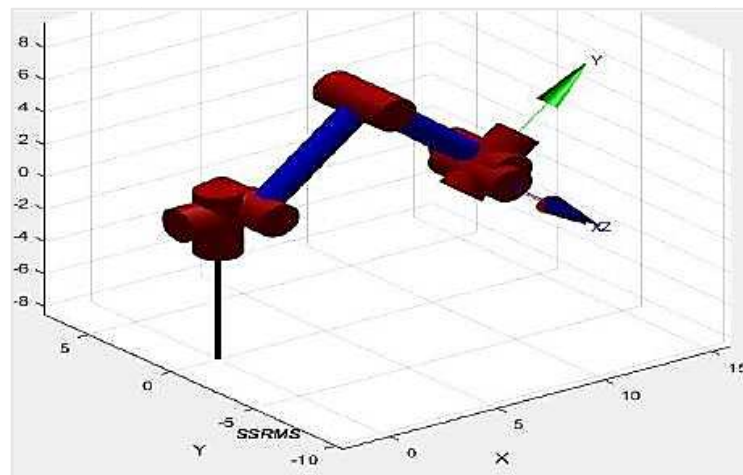
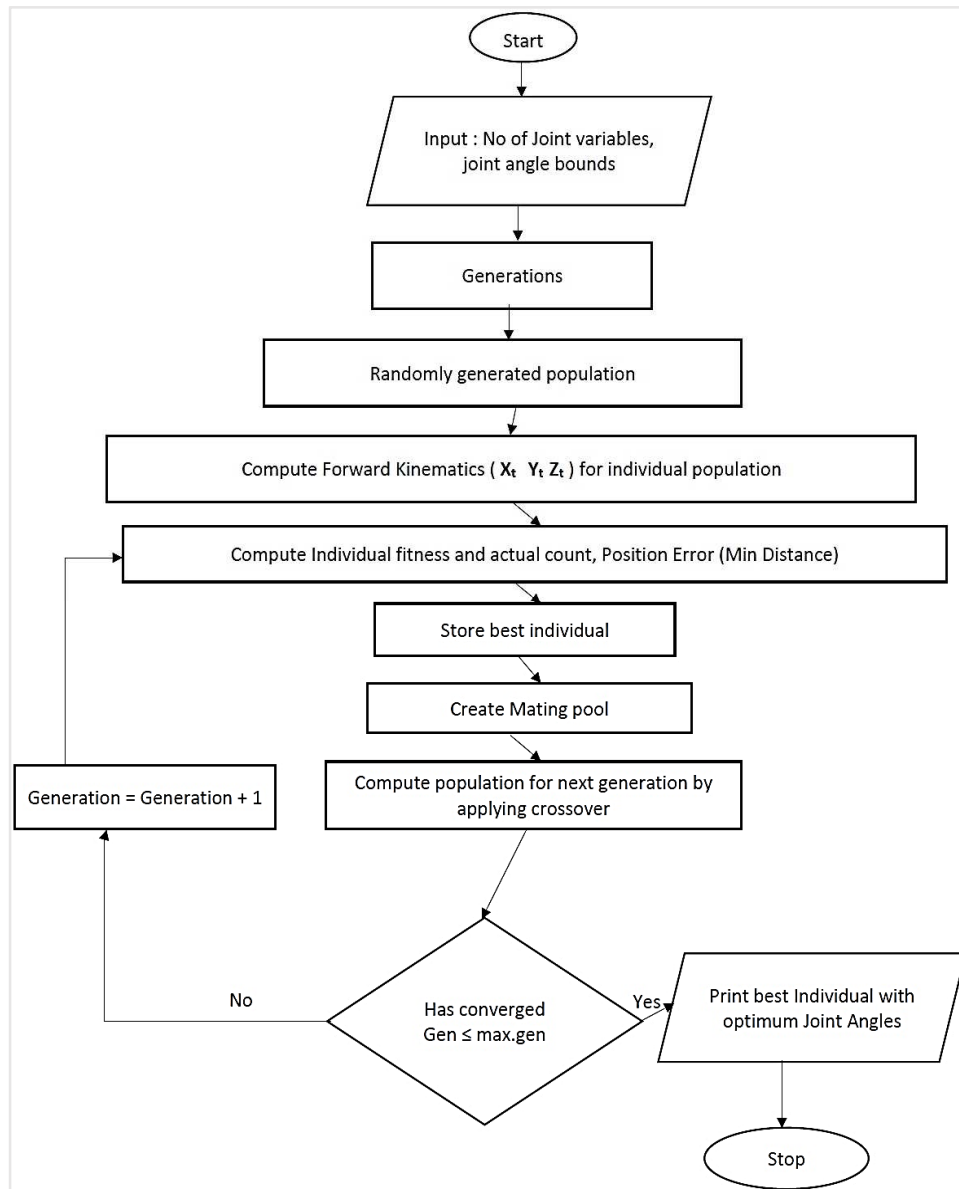


Figure5: Graphical Robot Representation of SSRMS using Robotics Toolbox for MATLAB

### 5.1 GA Flow Chart for Finding SSRMS Joint Angles

GA flow chart for finding optimum joint angles for SSRMS robot configuration.



**Figure 6: Flowchart for Finding Optimum Joint Angles Using GA**

**Table 2: D-H Parameters of SSRMS (7DOF)**

Frame	$a_i$ [Link Length]	$\alpha_i$ [Link Twist]	$d_i$ [Link Offset]	$\theta_i$ [Joint Angle]
1	0	$-\alpha_1$	$d_1$	$\theta_1$
2	0	$-\alpha_2$	$d_2$	$\theta_2$
3	$a_3$	0	$d_3$	$\theta_3$
4	$a_4$	0	$d_4$	$\theta_4$
5	0	$\alpha_5$	$d_5$	$\theta_5$
6	0	$-\alpha_6$	$d_6$	$\theta_6$
7	0	0	$d_7$	$\theta_7$

## 5.2 Forward Kinematics Equations of SSRMS

$$X_t = a_4 c_4 (s_1 s_3 + c_1 c_2 c_3) - d_2 s_1 - d_6 (c_5 (c_4 (c_3 s_1 - c_1 c_2 s_3) - s_4 (s_1 s_3 + c_1 c_2 c_3)) - s_5 (c_4 (s_1 s_3 + c_1 c_2 c_3) + s_4 (c_3 s_1 - c_1 c_2 s_3))) - d_7 (s_6 (c_5 (c_4 (s_1 s_3 + c_1 c_2 c_3) + s_4 (c_3 s_1 - c_1 c_2 s_3)) + s_5 (c_4 (c_3 s_1 - c_1 c_2 s_3) - s_4 (s_1 s_3 + c_1 c_2 c_3))) + c_1 c_6 s_2) + a_4 s_4 (c_3 s_1 - c_1 c_2 s_3) - c_1 d_3 s_2 - c_1 d_4 s_2 - c_1 d_5 s_2 + a_3 s_1 s_3 + a_3 c_1 c_2 c_3;$$

$$Y_t = c_1 d_2 + d_7 (s_6 (c_5 (c_4 (c_1 s_3 - c_2 c_3 s_1) + s_4 (c_1 c_3 + c_2 s_1 s_3)) + s_5 (c_4 (c_1 c_3 + c_2 s_1 s_3) - s_4 (c_1 s_3 - c_2 c_3 s_1))) - c_6 s_1 s_2) + d_6 (c_5 (c_4 (c_1 c_3 + c_2 s_1 s_3) - s_4 (c_1 s_3 - c_2 c_3 s_1)) - s_5 (c_4 (c_1 s_3 - c_2 c_3 s_1) + s_4 (c_1 c_3 + c_2 s_1 s_3))) - a_4 c_4 (c_1 s_3 - c_2 c_3 s_1) - a_4 s_4 (c_1 c_3 + c_2 s_1 s_3) - a_3 c_1 s_3 - d_3 s_1 s_2 - d_4 s_1 s_2 - d_5 s_1 s_2 + a_3 c_2 c_3 s_1;$$

$$Z_t = d_1 - d_6 (c_5 (c_3 s_2 s_4 + c_4 s_2 s_3) + s_5 (c_3 c_4 s_2 - s_2 s_3 s_4)) - c_2 d_3 - c_2 d_4 - c_2 d_5 - d_7 (c_2 c_6 - s_6 (c_5 (c_3 c_4 s_2 - s_2 s_3 s_4) - s_5 (c_3 s_2 s_4 + c_4 s_2 s_3))) - a_3 c_3 s_2 + a_4 s_2 s_3 s_4 - a_4 c_3 c_4 s_2;$$

Here  $c_n = \cos(\theta_n)$  here  $n = 1$  to  $7$   
 $s_n = \sin(\theta_n)$  here  $n = 1$  to  $7$

Here  $a_3 = 7.110$ ,  $a_4 = 7.110$ ,  $d_1 = 0.380$ ,  $d_2 = 1.360$ ,  $d_3 = 0.570$ ,  $d_4 = 0.475$ ,  $d_5 = 0.570$ ,  $d_6 = 0.635$ ,

$d_7 = 0$ ,  $\alpha_1 = -\pi/2$ ,  $\alpha_2 = -\pi/2$ ,  $\alpha_3 = 0$ ,  $\alpha_4 = 0$ ,  $\alpha_5 = \pi/2$ ,  $\alpha_6 = -\pi/2$ ,  $\alpha_7 = 0$ .

## 5.3 Forward Kinematics Equations Written in Fitness Function(Fitfcn)of GA Solver

$$\begin{aligned} xt = & a4 * (\cos(\min(4))) * ((\sin(\min(1))) * (\sin(\min(3)))) + (\cos(\min(1))) \\ & * (\cos(\min(2))) * (\cos(\min(3)))) - d2 * (\sin(\min(1))) - d6 \\ & * ((\cos(\min(5))) * ((\cos(\min(4))) * ((\cos(\min(3))) * (\sin(\min(1))) \\ & - (\cos(\min(1))) * (\cos(\min(2))) * (\sin(\min(3)))) - (\sin(\min(4))) \\ & * ((\sin(\min(1))) * (\sin(\min(3))) + (\cos(\min(1))) * (\cos(\min(2))) \\ & * (\cos(\min(3)))) - (\sin(\min(5))) * ((\cos(\min(4))) * ((\sin(\min(1))) \\ & * (\sin(\min(3))) + (\cos(\min(1))) * (\cos(\min(2))) * (\cos(\min(3)))) \\ & + (\sin(\min(4))) * ((\cos(\min(3))) * (\sin(\min(1))) - (\cos(\min(1))) \\ & * (\cos(\min(2))) * (\sin(\min(3)))) - d7 * ((\sin(\min(6))) \\ & * ((\cos(\min(5))) * ((\cos(\min(4))) * ((\sin(\min(1))) * (\sin(\min(3))) \\ & + (\cos(\min(1))) * (\cos(\min(2))) * (\cos(\min(3)))) + (\sin(\min(4))) \\ & * ((\cos(\min(3))) * (\sin(\min(1))) - (\cos(\min(1))) * (\cos(\min(2))) \\ & * (\sin(\min(3)))) + (\sin(\min(5))) * ((\cos(\min(4))) * ((\cos(\min(3))) \\ & * (\sin(\min(1))) - (\cos(\min(1))) * (\cos(\min(2))) * (\sin(\min(3)))) \\ & - (\sin(\min(4))) * ((\sin(\min(1))) * (\sin(\min(3))) + (\cos(\min(1))) \\ & * (\cos(\min(2))) * (\cos(\min(3)))) + (\cos(\min(1))) * (\cos(\min(6))) \\ & * (\sin(\min(2)))) + a4 * (\sin(\min(4))) * (\cos(\min(3))) \\ & * (\sin(\min(1))) - (\cos(\min(1))) * (\cos(\min(2))) * (\cos(\min(3))) \\ & - (\cos(\min(1))) * d3 * (\sin(\min(2))) - (\cos(\min(1))) * d4 \\ & * (\sin(\min(2))) - (\cos(\min(1))) * d5 * (\sin(\min(2))) + a3 \\ & * (\sin(\min(1))) * (\sin(\min(3))) + a3 * (\cos(\min(1))) * (\cos(\min(2))) \\ & * (\cos(\min(3))); \end{aligned}$$

$$\begin{aligned} yt = & (\cos(\min(1))) * d2 + d7 * ((\sin(\min(6))) * ((\cos(\min(5))) * ((\cos(\min(4))) \\ & * ((\cos(\min(1))) * (\sin(\min(3))) - (\cos(\min(2))) * (\cos(\min(3))) \\ & * (\sin(\min(1))) + (\sin(\min(4))) * ((\cos(\min(1))) * (\cos(\min(3))) \\ & + (\cos(\min(2))) * (\sin(\min(3)))) + (\sin(\min(5))) \\ & * ((\cos(\min(4))) * ((\cos(\min(1))) * (\cos(\min(3))) + (\cos(\min(2))) \\ & * (\sin(\min(1))) * (\sin(\min(3))) - (\sin(\min(4))) * ((\cos(\min(1))) \\ & * (\sin(\min(3))) - (\cos(\min(2))) * (\cos(\min(3))) * (\sin(\min(1)))) \\ & - (\cos(\min(6))) * (\sin(\min(1))) * (\sin(\min(2)))) + d6 * ((\cos(\min(5))) \\ & * ((\cos(\min(4))) * (\cos(\min(1))) * (\cos(\min(3))) + (\cos(\min(2))) \\ & * (\sin(\min(1))) * (\sin(\min(3))) - (\sin(\min(4))) * ((\cos(\min(1))) \\ & * (\sin(\min(3))) - (\cos(\min(2))) * (\cos(\min(3))) * (\sin(\min(1)))) \\ & - (\sin(\min(5))) * ((\cos(\min(4))) * ((\cos(\min(1))) * (\sin(\min(3))) \\ & - (\cos(\min(2))) * (\cos(\min(3))) * (\sin(\min(1)))) + (\sin(\min(4))) \\ & * ((\cos(\min(1))) * (\cos(\min(3))) + (\cos(\min(2))) * (\sin(\min(1))) \\ & * (\sin(\min(3)))) - a4 * (\cos(\min(4))) * ((\cos(\min(1))) * (\sin(\min(3))) \\ & - (\cos(\min(2))) * (\cos(\min(3))) * (\sin(\min(1)))) - a4 * (\sin(\min(4))) \\ & * ((\cos(\min(1))) * (\cos(\min(3))) + (\cos(\min(2))) * (\sin(\min(1))) \\ & * (\sin(\min(3))) - a3 * (\cos(\min(1))) * (\sin(\min(3))) - d3 * (\sin(\min(1))) \\ & * (\sin(\min(2))) - d4 * (\sin(\min(1))) * (\sin(\min(2))) - d5 * (\sin(\min(1))) \\ & * (\sin(\min(2))) + a3 * (\cos(\min(2))) * (\cos(\min(3))) * (\sin(\min(1))); \end{aligned}$$



$$\begin{aligned}
zt = & d1 - d6 * ((\cos(\min(5))) * ((\cos(\min(3))) * (\sin(\min(2))) * (\sin(\min(4))) \\
& + (\cos(\min(4))) * (\sin(\min(2))) * (\sin(\min(3)))) + (\sin(\min(5))) \\
& * ((\cos(\min(3))) * (\cos(\min(4))) * (\sin(\min(2))) - (\sin(\min(2))) \\
& * (\sin(\min(3))) * (\sin(\min(4)))) - (\cos(\min(2))) * d3 - (\cos(\min(2))) \\
& * d4 - (\cos(\min(2))) * d5 - d7 * ((\cos(\min(2))) * (\cos(\min(6))) \\
& - (\sin(\min(6))) * ((\cos(\min(5))) * ((\cos(\min(3))) * (\cos(\min(4))) \\
& * (\sin(\min(2))) - (\sin(\min(2))) * (\sin(\min(3))) * (\sin(\min(4))) ) \\
& - (\sin(\min(5))) * ((\cos(\min(3))) * (\sin(\min(2))) * (\sin(\min(4))) \\
& + (\cos(\min(4))) * (\sin(\min(2))) * (\sin(\min(3)))) - a3 * (\cos(\min(3))) \\
& * (\sin(\min(2))) + a4 * (\sin(\min(2))) * (\sin(\min(3))) * (\sin(\min(4))) \\
& - a4 * (\cos(\min(3))) * (\cos(\min(4))) * (\sin(\min(2)))));
\end{aligned}$$

## 5.4 Positioning Error

The ordinary straight-line distance between two points (Euclidean distance), in this case the distance between target point and the known point in the 3-D space, so the positioning error is given as,

$$P_{min} = \sum_{j=1}^n \sqrt{(x_i - p_x)^2 + (y_i - p_y)^2 + (z_i - p_z)^2}$$

GA solver with required input parameters and return values.

$$[\text{theta}, \text{fval}, \text{flag}, \text{output}] = \text{ga}(\text{Fitfcn}, \text{nvars}, [], [], [], [], \text{LB}, \text{UB}, [], \text{options});$$

## 6. RESULTS

Given task points.

$$P_1 = [p_x = 1.25; p_y = 2.32; p_z = 2.41;]$$

$$P_2 = [p_x = 1.38; p_y = 2.22; p_z = 1.21;]$$

As there are 7 unknown joint angles, each joint angle is considered to be of 4-bit length, Initial GA population has  $7 \times 4 = 28$  genes. GA will apply Selection, Crossover and Mutation operators on 28 binary strings (chromosomes) for creating the next generation. With the given set values, the chromosomes are randomly changed, and GA will generate better joint angles in every generation, which refers to reaching given task points by minimizing the positioning error and increased accuracy.

**Table 3: Joint Angles for given Task Points**

S.No	$\theta_1$ [rad]	$\theta_2$ [rad]	$\theta_3$ [rad]	$\theta_4$ [rad]	$\theta_5$ [rad]	$\theta_6$ [rad]	$\theta_7$ [rad]
P <sub>1</sub>	2.453	0.187	-0.149	2.449	0.263	-1.212	2.212
P <sub>2</sub>	0.724	0.334	0.424	1.342	-0.664	-0.661	0.617



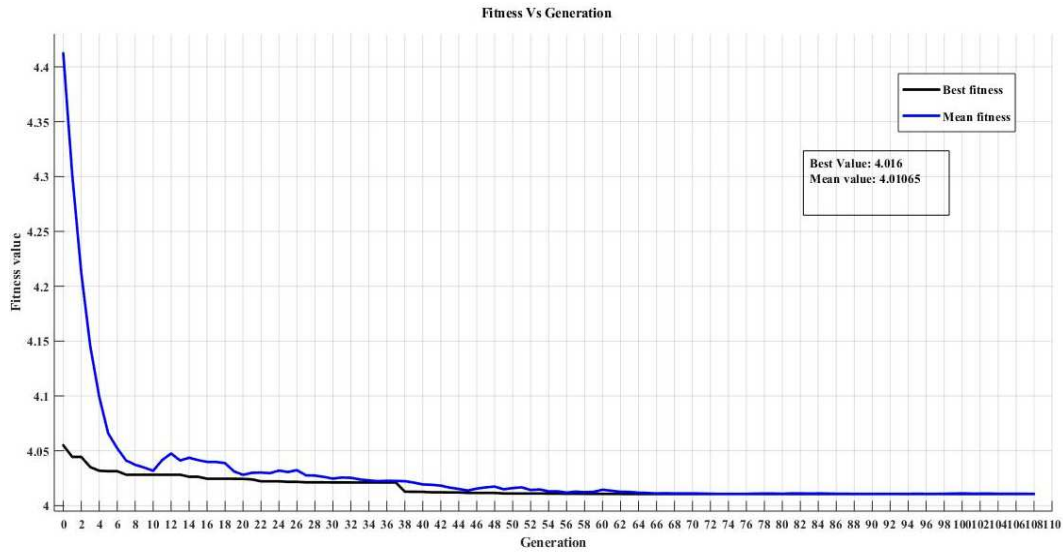


Figure 7: Fitness Vs Generation (Objective Function Evaluation for Joint Angles)

Table 4: GA Options (Set Values)

S.No	(GA) Set Values	Value
1	Population	200
2	Mutation Rate	0.01
3	Crossover fraction	0.8
4	Elite Count	$0.05 \times \text{Population}$
5	LB	Joint angle lower bounds
6	UB	Joint angle upper bounds
7	nvars	7
8	Options	Graphs(Fitness Vs Generation) etc.
9	Fitfcn	Fitness/objective function

Table 5: GA OUTPUT(Get Values)

S.No	OUTPUT(GA) Get Values	Value
1	Generations	51
2	Function-count	10400
3	Best Fitness	4.016
4	Mean Fitness	4.01065

## 7. CONCLUSIONS

In this paper the GA and TSP combinatorial optimizations are used to find the joint angles of SSRMS. The output (joint angles) returned by GA from giving operating points ( $P_1$ ,  $P_2$ ) is accurate, when the Upper bound (UB) and Lower bound (LB) is set on a heuristic basis. The time taken for solving 1 set of joints is 0.9 seconds, when the search space is not constrained with LB and UB on joint angles the time taken is 25 minutes. Joint angles obtained from GA are used with Robotics Toolbox for MATLAB for visualization of SSRMS robot. Finding Optimized trajectory for a given task is in the future scope of this work.

## 8. ACKNOWLEDGEMENTS

The writers of this manuscript are grateful to the support of (FIST sponsored Advance Prototyping and Manufacturing Lab) (SR/FST/ETI - 317/2012(C)) Department of Mechanical Engineering, K L E F (Deemed to be University) for supporting the writers of this manuscript throughout the project.

## REFERENCES

1. Wang, Hanlei, and YongchunXie. "Prediction error based adaptive Jacobian tracking for free-floating space manipulators." *IEEE Transactions on Aerospace and Electronic Systems* 48, no. 4 (2012): 3207-3221.
2. Ma, Ou, Hung Dang, and Khanh Pham. "On-orbit identification of inertia properties of spacecraft using a robotic arm." *Journal of guidance, control, and dynamics* 31, no. 6 (2008): 1761-1771.
3. Yoshida, Kazuya, Hiroki Nakanishi, Hiroshi Ueno, Noriyasu Inaba, Takeshi Nishimaki, and Mitsushige Oda. "Dynamics, control and impedance matching for robotic capture of a non-cooperative satellite." *Advanced Robotics* 18, no. 2 (2004): 175-198.
4. Rembala, Richard, and Cameron Ower. "Robotic assembly and maintenance of future space stations based on the ISS mission operations experience." *Acta Astronautica* 65, no. 7-8 (2009): 912-920.
5. Sabatini, Marco, Paolo Gasbarri, Riccardo Monti, and Giovanni Battista Palmerini. "Vibration control of a flexible space manipulator during on orbit operations." *Acta Astronautica* 73 (2012): 109-121.
6. Laryssa, Patten, Evans Lindsay, OshinowoLayi, Ochisor Marius, Kazuharu Nara, Lodewijk Aris, and T. Ed. "International space station robotics: a comparative study of ERA, JEMRMS and MSS." In *Proceedings of the 7th ESA Workshop on Advanced Space Technologies for Robotics and Automation*, Noordwijk, The Netherlands, pp. 1-8. 2002.
7. Figure 2. "Link(i) connected to joint (i) and joint (i + 1)" and Figure 3. "CAD model of SSRMS with 7 joints and DH-frames(Base to End-Effector)", [adapted] from Jazar, Reza N. *Theory of applied robotics: kinematics, dynamics, and control*. Springer Science & Business Media, 2010.
8. Holland, John H. "Genetic algorithms." *Scientific American* 267, no. 1 (1992): 66-73.
9. Corke, Peter. "Robotics toolbox for MATLAB, release 9 [Software]." *Robotics toolbox for MATLAB* (2011).
10. Xu, Wenfu, Yu She, and Yangsheng Xu. "Analytical and semi-analytical inverse kinematics of SSRMS-type manipulators with single joint locked failure." *Acta Astronautica* 105, no. 1 (2014): 201-217.
11. Števo, Stanislav, Ivan Sekaj, and Martin Dekan. "Optimization of robotic arm trajectory using genetic algorithm." *IFAC Proceedings Volumes* 47, no. 3 (2014): 1748-1753.
12. Moore, Randall C. "Image Analysis Overview." (2018).